Efficient image-based tracking of apparently changing moving targets

McIntyre, James; Church, Andrew; Labrosse, Frédéric

Publication date:
2009

Citation for published version (APA):
McIntyre, J., Church, A., & Labrosse, F. (2009). *Efficient image-based tracking of apparently changing moving targets*. http://hdl.handle.net/2160/3225

Efficient image-based tracking of apparently changing moving targets

James McIntyre, Andrew Church and Frédéric Labrosse

Computer Science department
Aberystwyth University
United Kingdom

e-mail: ffl@aber.ac.uk

Published in: *Proceedings of Towards Autonomous Robotics Systems*, pages 119–126 (2009), University of Ulster, Lodonderry, United Kingdom.

Efficient image-based tracking of apparently changing moving targets

James McIntyre Andrew Church Frédéric Labrosse
Computer Science department
Aberystwyth University
United Kingdom
[ffl@aber.ac.uk (F.L.)]

***Abstract*— In this paper, we present an efficient way of representing and tracking a moving object in images. In our approach, the object is visually represented as a set of pixels corresponding to an ideal view of the object as seen by a camera. As the object moves, its appearance can change in a number of ways, depending on the application. In this paper, we present two applications: leader-follower formation and visual guidance from a "camera in the sky". In these two applications, the object translates in the column and row directions of the images as well as, respectively, changes in size and orientation. However, other transformations, such as skewing and shearing, could be used in the proposed framework. We present results of real experiments performed in our Lab and show that even on low specification computers, the method performs well and fast enough.**

I. Introduction

The problem of tracking moving objects in images grabbed by a static or moving camera and controlling the camera and/or the tracked object to a desired relative pose has been the subject of many publications in the past. However, in most cases this has used explicit features of the objects that therefore need to be extracted from the images and tracked across the sequence of images. This is for example the case in [1] and [2] where a specific coloured target is fitted on the object to be tracked and the known colour explicitly extracted from the images. In [3], a target is painted on the object to be tracked, while in [4] a geometrical model of the object itself is used. These methods track specific features that make up the geometric model and the geometric transformation of the model is used to estimate the (pseudo) 3D pose of the object.

Another class of method uses the independent motion of the object. For example, in [5] the optical flow in omni-directional images is segmented to extract the moving objects and their motion properties are used to control the tracking. In [6], independent motion in stabilised images is extracted to create blobs that are then tracked using "appearance and shape similarity score".

In [7] a correlation filter-based comparison of the object and current views is used, using optimal filters that are adapted to accommodate for 3D variations of the object. This method is appealing but requires the non-obvious step of determining the filters. Moreover, adapting the filters to cope with visual changes of the object could lead to drift in the used filter that could result in detection failure.

In [8], a comparison between Image-Based Visual Servoing (IVBS) and Position-Based Visual Servoing (PVBS) is made and concludes that IVBS could lead to more stable servoing control compared to PBVS. However, this also concludes that if the starting point is far from the destination, a less than optimum behaviour could be produced. It is clear that if optimal control is desired, the camera must be calibrated and its relationship to the space where the object is controlled must be taken into account. For example, in the three very similar papers [9], [10], [11], differential flatness is used to linearise the relationship between image and ground control. Recently, such calibration has been used to perform servoing based on the tracking of features in a pinhole [12] and omni-directional [13] camera using geometrical constraints of the motion of the camera and its geometry.

In this paper, we use a different approach where the object is represented by its appearance, i.e., an image of it, as in [14]. The appearance is in fact a set of pixels the position of which can be transformed to take into account changes in the apparent shape of the object.

The main contribution of this paper is the presentation of a framework that allows the efficient localisation of the appearance of the object in an image even after transformations that would normally require re-sampling of the images. This framework is applied to two different applications, showing its usefulness and applicability. The applications are:

- leader-follower formation: the follower robot visually tracks and follows an object only using an image of it;
- visual guidance of a robot on the ground from a camera in the sky: a camera overlooking an area on the ground on which a robot can move is used to track and control the robot to specified positions in the image.

The two applications and their experimental results are given in Sections III and IV. Section II presents the framework, including the appearance representation and recognition, the method used to track the appearance in the succession of images and how this can be used to control the camera and/or the object (if controllable) to adjust their relative pose. Section V discusses the proposed methods and offers a conclusion.

II. Object specification and recognition

In previous work we have compared two entire images [15], [16], [17] or sub-images [14], [18] to evaluate changes in view point, track objects or register images. However, we have always assumed that either the images were not geometrically transformed or have precomputed these transformations (e.g., using warping in [17] or scaling in [14]). In this work, we propose to dynamically transform sub-images to cope with larger changes in view point. Indeed, many applications where tracking an object is required imply large

changes in camera-object relative pose, therefore implying important changes in appearance of the object. For example, in [14], the change was due to variable distance between the camera (carried by a follower robot) and the object (a leader robot). In this previous instance, the object was represented as a number of appearances corresponding to a number of distances, the change being reflected as a change of the size of the appearance of the leader.

Dynamically transforming sub-images is expensive. The problem lies in the fact that transforming the pixel positions results in an new image lattice that does not anymore align with the lattice of the image in which the sub-image must be located. The traditional method to resolve that problem is to re-sample the transformed image on the lattice of pixels that correspond to the un-transformed lattice by computing new pixel values from the surrounding pixels. This can be done using bi-linear interpolation or, a faster method, nearest neighbour pixel value. Note that more expensive methods do exist that will provide visually better results. However, whichever interpolation method is used, no new information is created from the original sub-image. In fact, in the case of a scale reduction, information is even lost.

This therefore results in un-necessary processing when the only reason for the creation of the new re-aligned sub-image is to compare it with another image.

A. Comparing images

Instead of re-sampling the transformed sub-image on the original lattice, we propose to only use the original pixels (after transformation) for the image comparison. Indeed, since the re-sampling does not introduce new information, at best, only using the original pixel values should provide a similar result.

As in previous work, we consider images (or sub-images) as points in the image space and use the Euclidean distance to compare them. More formally, we define $\mathbf{p} \in \mathbb{R}^2$ as a pixel position. Note that positions are not discrete. However, we define the operator $\eta(\cdot)$ that returns the nearest neighbour discrete position of a continuous one: $\eta(\mathbf{p}) \in \mathbb{Z}^2$. Also note that coordinates outside the image area will have to be treated in some way, depending on the application. This is done by the operator $\iota(\cdot)$: $\iota(\mathbf{p}) \in \mathbb{P}$, where $\mathbb{P}$ is the set of pixel positions delimited by the image size: $\mathbb{P} = \{(x, y) \mid x \in \mathbb{R}, 0 \leq x \leq w, y \in \mathbb{R}, 0 \leq y \leq h\}$, where w and h are respectively the width and height of the images.

The operator $\Theta(\cdot)$ geometrically transforms positions in two dimensions: $\Theta(\mathbf{p}) \in \mathbb{R}^2$. Depending on the application, this operator can correspond to, but is not limited to, a combination of translation, scaling and rotation.

Associated to a position in an image is a pixel value $\mathbf{v}$ that takes its values in some space $\mathbb{C}$ that depends on the actual colour space used. In all the experiments reported here we used the Red-Green-Blue (RGB) colour space and $\mathbb{C}$ is thus a space of dimensionality 3 and $\mathbf{v}$ is a three-component vector of integer values between 0 and 255: $\mathbf{v} = ({}^{R}v \quad {}^{G}v \quad {}^{B}v)^{\mathrm{T}}$.

The appearance $\mathcal{A}$ of an object is thus a set of pairs of positions and pixel values:

$$\mathcal{A} = \{(\mathbf{p}_i, \mathbf{v}_i) \mid \mathbf{p}_i \in \mathbb{R}^2, \mathbf{v}_i \in \mathbb{C}\}.$$

Similarly, an image $\mathcal{I}$ is a set of pairs of discrete pixel positions and corresponding pixel values:

$$\mathcal{I} = \{(\mathbf{q}_i, \mathbf{v}_i) \mid \mathbf{q}_i \in \mathbb{P}, \mathbf{v}_i \in \mathbb{C}\}.$$

For ease of notation, we will use the same symbol as an operator returning a specified colour component k of the pixel value of a given discrete pixel position:

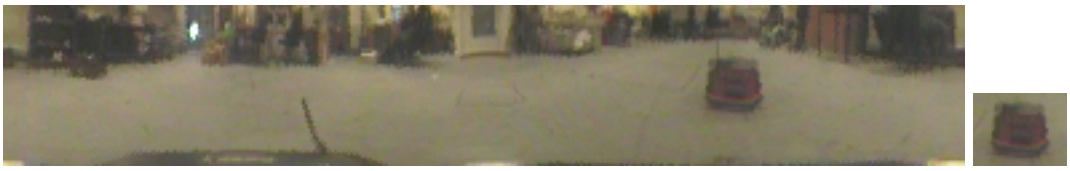
$${}^{k}v_i = \mathcal{I}(\mathbf{q}_i, k).$$

We can now transform an appearance and compare the result with an image using the Euclidean distance:

$$d(\mathcal{I}, \mathcal{A}, \Theta) = \sqrt{\sum_{i=1}^{|\mathcal{A}|} \sum_{k=1}^{c} \left\{ \mathcal{I}\left(\eta\left(\iota\left(\Theta\left(\mathbf{p}_i \right) \right) \right), k \right) - {}^{k}v_i \right\}^2}, \tag{1}$$

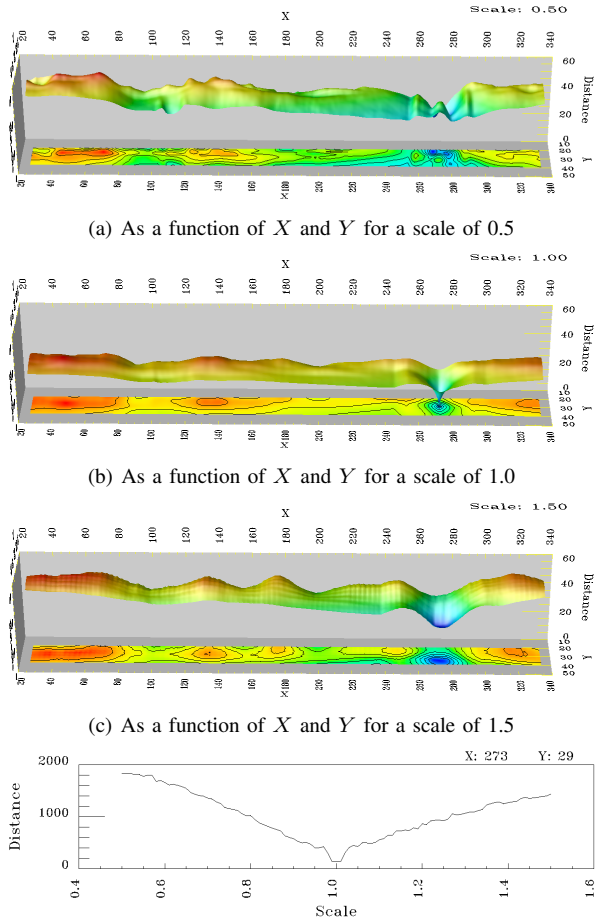
where $(\mathbf{p}_i, \mathbf{v}_i) \in \mathcal{A}$, $|\mathcal{A}|$ is the number of pixels in the appearance and c is the number of colour components. The effect of computing this distance is to compare all the pixels of the appearance $\mathcal{A}$, after they have been transformed using $\Theta(\cdot)$, to the pixels of the image $\mathcal{I}$ that fall near the transformed appearance pixels. The lower the distance is, the better the match is between the transformed appearance with the corresponding part of the current image. A distance of 0 would indicate a perfect match. Figure 1 shows the distance function (1) between the image and appearance shown in Figure 2 for a transformation that represents a scaling (multiplication of the size by a given factor) of the appearance and a horizontal (X) and vertical (Y) translation. Figure 1(b) shows a cone around the position that corresponds to where the object is, the bottom of which is easy to locate. However, when the object's appearance is scaled, the distance function presents steps (Figure 1(d)) and a grid pattern (Figure 1(c)) that are due to the non-explicit re-interpolation of the appearance (a phenomenon observed in multi-scale image registration [18]).

From an implementation point of view, computing the distance value (1) can be done efficiently, especially when compared to actually re-sampling the appearance. Only the positions of the pixels are transformed and no new pixel value is created. When the distance is computed, the positions are at least moved to the nearest discrete neighbour (using the operator $\eta(\cdot)$, usually a rounding operation) and maybe modified to fit the image (operator $\iota(\cdot)$), operations that are typically fast (e.g., clamping or modulo). Moreover, since no new pixels are created, the computation of the distance value is independent of the transformation. For example, if the transformation involves scaling of the appearance and if a re-sampling was performed, then the distance computation would vary depending on the scaling. Such variable computation time is generally not good when a Proportional-Integral-Derivative (PID) controller is used (see Section II-C) and keeping it constant is therefore good.



(a) (b)

Fig. 2. A typical image (a) and appearance (b) for the leader-follower application



(a) As a function of X and Y for a scale of 0.5

(b) As a function of X and Y for a scale of 1.0

(c) As a function of X and Y for a scale of 1.5

(d) As a function of scale for the translation that gives the minimum distance

Fig. 1. Distance function for a transformation including horizontal (X) and vertical (Y) translations and scaling of the size of the appearance

B. Object tracking

The problem of finding the object given its appearance $\mathcal{A}$ in the current image $\mathcal{I}_c$ is now one of minimising the distance in (1) as a function of the transformation Θ. We therefore obtain the current transformation Θ_c as:

$$\Theta_c = \arg\min_{\Theta}(d(\mathcal{I}_c, \mathcal{A}, \Theta)). \quad (2)$$

When there is no information as to the value of the transformation, a global search must be performed. In most cases however, some assumptions can be made and a guided global, or a semi exhaustive, search can be used.

In most cases however, the object can be tracked from a previously known transformation. This is done using a local minimisation method. Using a gradient descent method would be tempting. However, such a method needs the gradient of the function, which involves making additional evaluations of the distance function. More importantly however, and as can be seen in Figure 1, the gradient can locally vanish even for transformations that are not at a minimum of the function. This is because ultimately, the pixels of the appearance are compared with pixels of the current image at discrete pixel positions. This, depending on the transformation, can create flat areas of the distance function corresponding to appearance pixels lying in between discrete positions of the image.

As a result, we use the simplex algorithm of Nelder and Mead as implemented in the Gnu Scientific Library (GSL)[1]. The method uses a simplex with $n + 1$ vertices, where n is the dimensionality of the function to minimise, and applies geometrical transformations to the simplex such as reflections, expansions and contractions, to try to improve the function value at the vertices of the simplex. Using a succession of these transformations, the simplex moves through the parameter space converging towards the minimum of the function where it contracts to a small size. A threshold on the size is used to stop the minimisation and is indicative of the precision reached.

C. Control loop

In applications where some image stabilisation is needed, a desired transformation is available. The problem is then to modify the system, e.g., by moving the camera, so that the current transformation converges towards the desired transformation. This is the so-called Image-Based Visual Servoing control as described in [8].

The actual control technique depends on the application but generally a PID controller is adequate, the error used being the difference between the current and desired transformations. This is discussed further in Sections III and IV for the two presented applications.

III. "Follow that object"

A. Description of the application

The first application is an implementation of the classic leader-follower problem. We presented a solution to that problem in a previous publication [14], which had a number of problems, most of them being solved here.

In the leader-follower problem, a moving object is being tracked in images captured by a camera mounted on a robot

[1]GSL: `http://www.gnu.org/software/gsl/`

and the task is to control the robot so that the object remains at a given position and scale in the sequence of images.

For this work, we used a panoramic camera that uses a hyperbolic mirror to produce panoramic images that are 360 pixels wide (one pixel per degree of angle) and 50 pixels high (see [16] for a complete description of the system), Figure 2. The distortions introduced by the camera are not corrected. Therefore, the object will be non-linearly distorted as it moves further away from the camera, depending on the position of the camera. Indeed, if the camera is above the object, as is the case here, the object will be reflected on different parts of the mirror to project on the image, which, because of its non-linear curvature changes as a function of height from the tip, will result in a non-linear scaling. Moreover, the object will move up and down in the image as a function of the distance camera-object[2] in a non-linear way. However, the experiments show that not compensating for these distortions is not needed.

For this application, the transformation $\Theta(\cdot)$ contains two terms: the first is a scaling of the appearance (multiplication of the coordinates of the pixels of the appearance by a factor); the second is a translation along the two axes of the image (addition of a 2D vector to the coordinates). The scaling results in moving the pixels apart (for scaling factors greater than 1.0) or closer to each other (for scaling factors lower than 1.0). This has the effect of simulating a change in size of the object due to a change in camera-object distance. The horizontal translation simulates the effect of the object moving sideways in the field of view while the vertical translation, combined with the scaling, simulates the camera-object distance changing. The minimisation in (2) was therefore performed against these three parameters. The initial transformation was determined by a first limited global search and then by a local minimisation using the simplex method (Section II-B).

The operator $\iota(\cdot)$ is in this case a clamping of the vertical pixel position to the limit of the vertical range and a modulo of the horizontal position to implement the wrapping around of the panoramic image.

The control uses two of the transformation parameters. The horizontal position is compared to the ideal horizontal position that corresponds to the front of the robot while the scaling factor is compared to the ideal factor, i.e., the one that sets the correct camera-object distance (1.0 in all the experiments reported here). The difference between the actual and ideal parameters is used in a PID controller setting the turning rate (from the horizontal position error) and speed (from the scaling error) with parameters manually determined.

B. Experiments and results

To evaluate the performance of the follower, we used the appearance shown in Figure 2(b) and teleoperated the leader robot along a variety of trajectories, repeating each ten times. The positions of the leader and follower were recorded using our Vicon 512 motion tracking system and for each of the experiments we present a typical trajectory for both robots, statistics on the leader-follower distance for the ten runs and the graphs of the distances for all ten runs. This was done for a STRAIGHTLINE, an OVAL and a FIGUREOFEIGHT, Figures 3 to 8 and Tables I to III.

[2]This fact was exploited in [14] to estimate the distance camera-object.

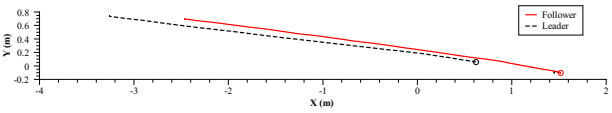


Fig. 3. STRAIGHTLINE: typical trajectories

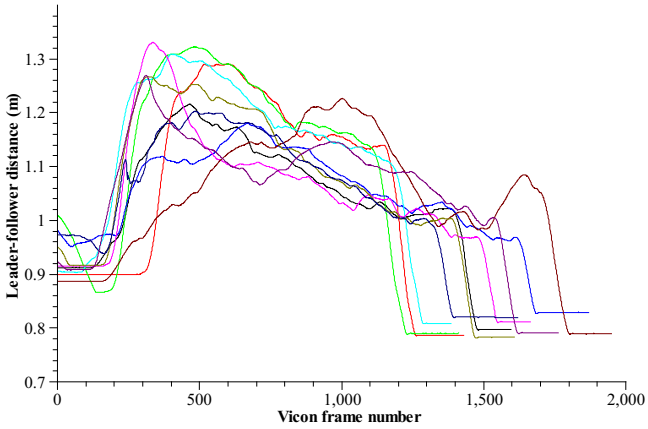


Fig. 4. STRAIGHTLINE: leader-follower distances

In all these experiments, the desired leader-follower distance was 1 m and the tracking was performed at approximately 12 frames per second. The software was executed on the on-board computer (a Pentium III at 800 MHz).

It is clear that in all cases, the performance is good with actual leader-follower distance close to the desired value and small standard deviation. This is a good improvement over the previously published method [14], in part due to the use of a PID controller instead of a simple P controller. Note however that the parameters of the PID controller could have been better fine tuned to obtain an even better performance.

The speed of the algorithm is also good, much better than the previously published one (12 fps instead of 4 fps on the same hardware).

Finally, this new method is better than the previously presented one in that it does not make any assumptions as to the geometry of the system while we previously assumed that the camera was above the leader and that the floor was flat. These two assumptions are not needed anymore as their effect is now explicitly taken into account by the transformation of the appearance of the leader.

IV. "THE EYE IN THE SKY"

A. Description of the application

The second application involves a different transformation. The application is to track and guide a mobile robot on the ground seen from a "camera in the sky". The results presented here were using a camera mounted on the ceiling

(a) (b)

Fig. 9. A typical image from the sky (a) and appearance (b)

TABLE I
STRAIGHTLINE: DISTANCE STATISTICS (IN M)

Index	Min	Max	Mean	Std. dev.
1	0.80	1.22	1.04	0.11
2	0.79	1.29	1.06	0.17
3	0.79	1.32	1.10	0.18
4	0.83	1.18	1.03	0.10
5	0.81	1.31	1.12	0.15
6	0.81	1.33	1.04	0.12
7	0.79	1.23	1.03	0.12
8	0.78	1.27	1.06	0.14
9	0.82	1.20	1.04	0.12
10	0.79	1.27	1.05	0.12

TABLE II
OVAL: DISTANCE STATISTICS (IN M)

Index	Min	Max	Mean	Std. dev.
1	0.86	1.22	1.09	0.09
2	0.73	1.46	1.12	0.18
3	0.77	1.47	1.16	0.17
4	0.75	1.35	1.10	0.14
5	0.76	1.39	1.15	0.14
6	0.80	1.24	1.06	0.14
7	0.81	1.53	1.21	0.17
8	0.75	1.49	1.16	0.20
9	0.74	1.40	1.13	0.14
10	0.73	1.36	1.10	0.16

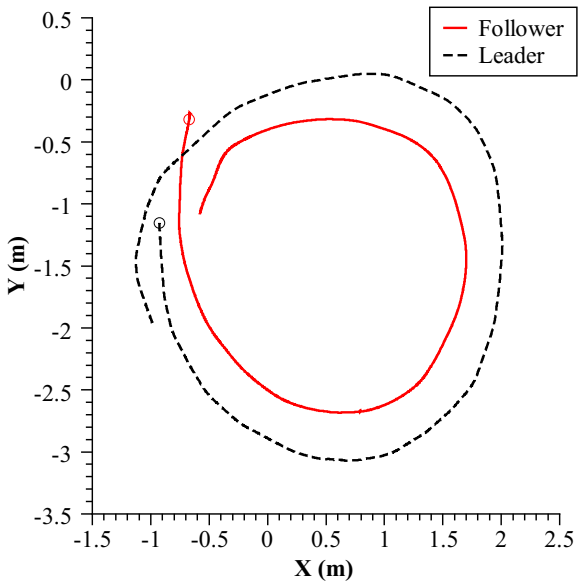


Fig. 5. OVAL: typical trajectories

of our Lab. However, the goal is to use a camera mounted on a flying platform such as a kite [19], [20]. In this application, an image of the robot as seen in the camera is given to the system and constitutes the appearance of the robot to be tracked. The destination of the robot is given as a position in the image plane by a person clicking on the live image displayed in a Graphical User Interface (GUI). The robot is then tracked and directed to its destination. The experiments reported here took place in our Lab, on grey carpet using a black topped Pioneer robot. To make the robot, and particularly its orientation, more visually prominent, a

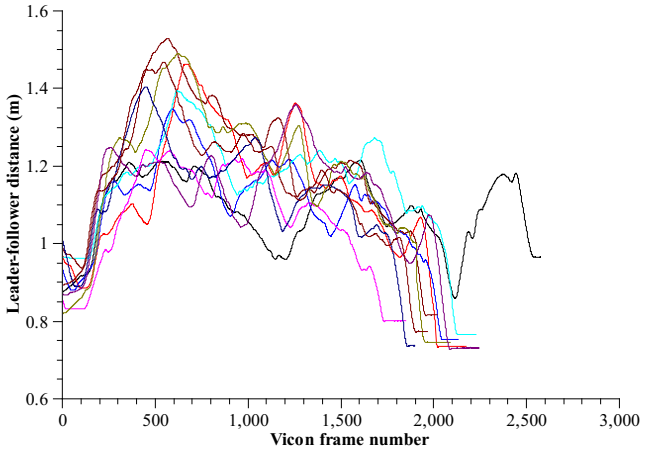


Fig. 6. OVAL: leader-follower distances

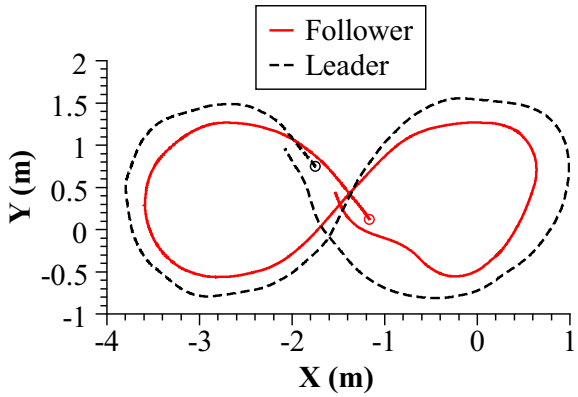


Fig. 7. FIGUREOFEIGHT: typical trajectories

set of coloured ribbons arranged in a triangle was set on top of the robot (Figure 9).

The transformation $\Theta(\cdot)$ again contains a translation term (horizontal and vertical in the image). The other term is a rotation that rotates the appearance around its centre. Note that the way the robot looks will not remain constant as it moves around in the field of view of the camera. This is because even if the camera's optical axis is vertical, the change in view point exposes different parts of the robot. Moreover, un-corrected distortions introduced by the wide-angle lens used for the experiments reported here imply that the shape of the robot changes as it moves in the field of view of the camera. This could easily be solved by calibrating the camera. Finally, the mapping between pixel coordinates and positions on the ground is not constant because the line of sight to the robot is not always vertical. Using differential flatness [9], this could be taken into account, provided the pose of the camera is known, and the ground assumed to be flat, which is unlikely given our goal of the camera in the sky. Therefore, no calibration of the camera or correction for the position of the robot was taken into account.

For this application, the operator $\iota(\cdot)$ was restricted to a clamping of the appearance to the size of the image.

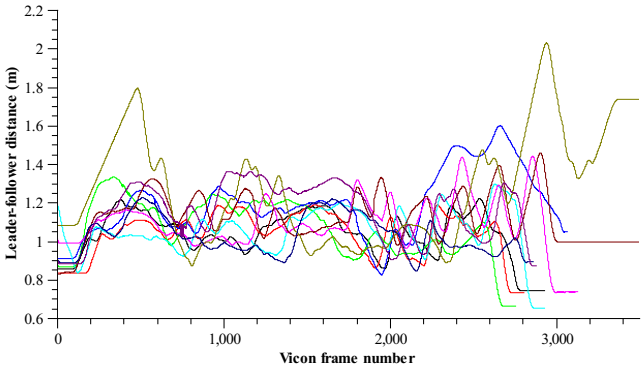


Fig. 8. FIGUREOFEIGHT: leader-follower distances

TABLE III
FIGUREOFEIGHT: DISTANCE STATISTICS (IN M)

Index	Min	Max	Mean	Std. dev.
1	0.75	1.22	1.04	0.12
2	0.73	1.19	1.04	0.11
3	0.67	1.33	1.07	0.15
4	0.83	1.60	1.19	0.17
5	0.65	1.30	1.05	0.12
6	0.74	1.44	1.09	0.13
7	0.83	1.46	1.05	0.10
8	0.87	2.03	1.32	0.30
9	0.85	1.23	1.03	0.10
10	0.87	1.36	1.16	0.14

The first pose of the robot was found by doing a limited global minimisation of (2) (only a subset of all possible transformation values was considered). Once the robot was acquired, a local search using the simplex method was used (Section II-B).

A proportional controller was used to minimise the error between the detected position and the desired position of the robot, with a threshold on the speed. A change in orientation of the robot was computed from its current orientation and the straight line to the destination. To minimise the space needed by the robot to reach its destination, the robot was first rotated on the spot to within 30° of the desired orientation. Then both orientation and position were set as a proportion of the error in position and orientation.

B. Experiments and results

To evaluate the performance of our object tracking method applied to this application, we have conducted three different experiments. The first one was designed to show the repeatability of the tracking by positioning the guided robot approximately at the same starting pose and guiding it to the same point on the image. Nine runs were recorded using our Vicon 512 motion tracking system and the trajectories are plotted in Figure 10. Clearly, repeatability is good since the largest deviation between trajectories is approximately 6 cm.

The next experiment is to show that the same position in the image can be reached from a variety of starting poses, Figure 11. Again, the tracking of the robot is good: all final positions are approximately within 10 cm of each other.

Finally, the robot was guided from the top left corner of the image to the bottom left of the image and vice-versa, twice

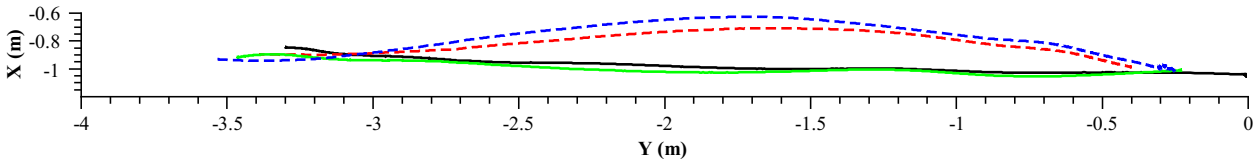


Fig. 12. Straight lines in the image from the top left to bottom left of the image (continuous lines) and vice-versa (dashed lines)

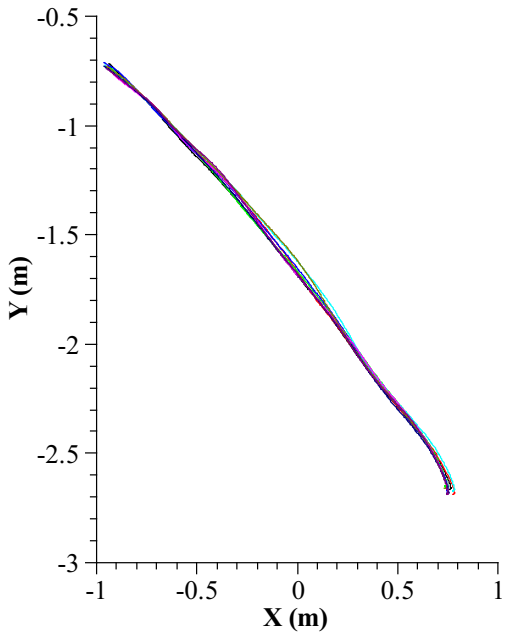


Fig. 10. Nine trajectories of the robot from approximately the same position on the floor to the same position in the image

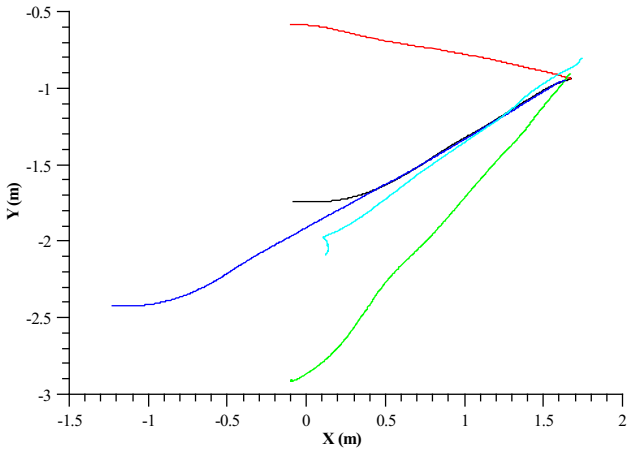


Fig. 11. Five trajectories from different poses to the same position in the image

in each direction, Figure 12. In this case, the target position was set each time and was therefore only approximately the same. However, the experiment shows an asymmetry in the performance. This is partly due to the distortions in the image due to the wide angle lens and the fact that the appearance of the object is not symmetric. This resulted in the orientation of the object being systematically slightly wrongly estimated when the robot was going upwards (but also to a lesser extent downwards) in the image. Moreover, it is not surprising that the robot should in fact follow a trajectory that is slightly biased towards the centre of the image given the non-corrected image distortions.

In this experiment too the frame rate was approximately 12 fps, although on a faster computer (Pentium IV at 2.8 GHz) than the one on-board the Pioneer. The effective slower frame rate is due to a decentralised application with networked image acquisition, remote image processing, larger appearance (the images in Figure 9 are at half scale) and a graphical user interface used to show the robot being tracked and set destinations for it. The application was designed as such so that all three aspects of it (image acquisition, image processing and robot control) could be run on different machines.

V. Discussion and Conclusion

The method we have presented to track moving objects was applied to two different applications. These involve different types of control of the platform holding the camera or of the robot being tracked. The method was shown to work well and to be efficient given the frame rate obtained on low specification computers without any optimisation of the code but that provided by the compiler. We have shown that transforming the appearance of the object to take into account apparent shape changes can be done efficiently.

Moreover, we have shown that IBVS can be done without camera calibration or knowledge about the relative pose of the camera and object. It is clear that this results in a non-constant effect of the control to the camera and/or object. For example, as can be seen in Figure 9, the distortions in the images introduced by the wide angle lens mean that the robot controlled along a straight line in the image will actually move along a curved trajectory on the ground. This could clearly be a problem if the robot had to follow straight lines. A solution to that problem could be to explicitly represent the apparent change in shape of the object in the transformation of its appearance to estimate its pseudo 3D pose. This could then be used in the control. For example, if the ground was not flat, this would result in a shortening of the appearance indicating that the robot is moving up or down a slope and that its speed in the image must be adapted if a constant speed on the ground is needed. This more complex transformation could also be used to obtain a better match of the object in the images, but would slow down the matching procedure. In fact, the only failures we have experienced with

the "camera in the sky" experiment happened when the robot was near the edges of the image, i.e., where the distortions in appearance are maximum. This problem was especially acute during the initial global search where the system was not recognising the object in such situations, but not during the tracking where the search is only local. A transformation including some way of shortening the appearance could have solved this problem.

Another area of possible improvement is in tackling changes in colour of the target due to changes in lighting conditions. Such changes could be due to overall illumination changes, which could be tackled by using a more appropriate colour space; we used here RGB but CIE $L^*a^*b^*$ could solve that issue by not using the luminance information [21]. The changes could also be due to shadows and/or changes in colour cast due to different times of the day. The appearance could be adapted in a controlled fashion to take into account new pixel values from the successfully localised object. This could however result in the appearance changing so much that it would not correspond to the object anymore.

Finally, partial occlusion of the appearance could make the tracking fail. The minimisation of the distance (2) could be biased depending on what of the appearance is occluded and by what. This is essentially a failure of the RGB colour space where, for example, the distance red-black is lower than the distance yellow-black because red has only one component at 255 while yellow has two. This means that if a yellow part of the appearance is occluded by black this would result in a worse match than if it had been red occluded by black. However, changing the colour space alone will not solve this problem as all colour spaces share the problem of not all colours being equally different. Occlusions should therefore be taken into account in the metric used to compare images. This is still an open problem.

To conclude, we have presented a method that allows the efficient localisation and tracking of moving objects in images acquired by a possibly moving camera where the objects are only specified by a set of colours and their relative spatial position that can be transformed to take into account apparent changes of the shape of the object. We have applied the method to two different applications that involve two different types of transformation (scaling and rotation) that would normally require the expensive step of image re-sampling. Our method to compare the object's appearance to pixels in the current image does not lose information or does not attempt to falsely create more since all and only the pixels of the original appearance are used.

We have shown a good performance of controllers based on the tracking of the objects, both in terms of spatial accuracy and repeatability and processing efficiency. In addition to the experiments formally recorded and presented earlier we have extensively tested our software by tracking a range of objects in both applications, including following a human being using our large outdoor robot. In all cases, similar tracking performance was exhibited by our method.

REFERENCES

[1] S. Y. Chiem and E. Cervera, "Vision-based robot formations with bézier trajectories," in *Proceedings of the International Conference on Intelligent Robots and Systems*, 2004, pp. 191–198.

[2] G. Klančar, M. Brezak, D. Matko, and I. Petrović, "Mobile robots tracking using computer vision," *Automatika*, vol. 46, no. 3–4, pp. 155–163, 2005.

[3] H. Schneiderman, M. Nashman, A. J. Wavering, and R. Lumia, "Vision-based robotic convoy driving," *Machine Vision and Applications*, vol. 8, no. 6, pp. 359–364, 1995.

[4] E. Malis and F. Chaumette, "2 1/2 D visual servoing with respect to unknown objects through a new estimation scheme of camera displacement," *International Journal of Computer Vision*, vol. 37, no. 1, pp. 79–97, 2000.

[5] R. Vidal, O. Shakernia, and S. Sastry, "Formation control of nonholonomic mobile robots with omnidirectional visual servoing and motion segmentation," in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 1, 2003, pp. 584–589.

[6] S. Ali and M. Shah, "Cocoa: Tracking in aerial imagery," in *Airborne Intelligence, Surveillance, Reconnaissance (ISR) Systems and Applications III, Proceedings of SPIE*, ser. Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, vol. 6209, Orlando, FL, 2006.

[7] M. W. Eklund, G. Ravichandran, M. M. Trivedi, and S. B. Marapane, "Real-time visual tracking using correlation techniques," in *Proceedinds of the IEEE Workshop on Applications of Computer Vision*, 1994, pp. 256–263.

[8] F. Chaumette and S. Hutchinson, "Visual servo control, part I: Basic approaches," *IEEE Robotics and Automation Magazine*, vol. 13, no. 4, pp. 82–90, 2006.

[9] R. S. Rao, V. Kumar, and C. J. Taylor, "Visual servoing of a ugv from a uav using differential flatness," in *Proceedings of the International Conference on Intelligent Robots and Systems*, Las Vegas, NV, 2003, pp. 743–748.

[10] R. S. Rao, C. J. Taylor, and V. Kumar, "Calibrating an air-ground control system from motion correspondences," vol. 2, Washington, DC, 2004, pp. 218–225.

[11] R. S. Rao, V. Kumar, and C. J. Taylor, "Planning and control of mobile robots in image space from overhead cameras," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Barcelona, Spain, 2005, pp. 2185–2190.

[12] G. L. Mariottini and D. Oriolo, Guiseppeand Prattichizzo, "Image-based visual servoing for nonholonomic mobile robots using epipolar geometry," *IEEE Transactions on Robotics*, vol. 23, no. 1, pp. 87–100, 2007.

[13] G. L. Mariottini and D. Prattichizzo, "Image-based visual servoing with central catadioptric camera," *International Journal of Robotics Research*, vol. 27, pp. 41–57, 2008.

[14] J. Oliver and F. Labrosse, "Towards an appearance-based approach to the leader-follower formation problem," in *Proceedings of Towards Autonomous Robotic Systems*, 2007, pp. 137–144.

[15] M. Neal and F. Labrosse, "Rotation-invariant appearance based maps for robot navigation using an artificial immune network algorithm," in *Proceedings of the Congress on Evolutionary Computation*, vol. 1, Portland, Oregon, USA, 2004, pp. 863–870.

[16] F. Labrosse, "The visual compass: Performance and limitations of an appearance-based method," *Journal of Field Robotics*, vol. 23, no. 10, pp. 913–941, 2006.

[17] ——, "Short and long-range visual navigation using warped panoramic images," *Robotics and Autonomous Systems*, vol. 55, no. 9, pp. 675–684, 2007.

[18] P. Bunting, F. Labrosse, and R. Lucas, "A multi-resolution area-based technique for automatic multi-modal image registration," *Image and Vision Computing*, 2009, under revision after comments from reviewers.

[19] J. C. Murray, M. J. Neal, and F. Labrosse, "Intelligent kite aerial platform for site photography," in *Proceedings of the IEEE Conference on Automation Science and Engineering*, Scottsdale, AZ, 2007, pp. 553–558.

[20] H. Hosseini, M. Neal, and F. Labrosse, "Error surface generation techniques for appearance-based stabilisation of an intelligent kite aerial photography platform (ikapp)," in *Proceedings of Towards Autonomous Robotic Systems*, University of Edinburgh, UK, 2008, pp. 163–170.

[21] A. Woodland and F. Labrosse, "On the separation of luminance from colour in images," in *Proceedings of the International Conference on Vision, Video, and Graphics*, University of Edinburgh, UK, 2005, pp. 29–36.